A Measure of the Connection Strengths between Graph Vertices with Applications

Jie Chen*   Ilya Safro†

Abstract

   We present a simple iterative strategy for measuring the connection strength between a pair of vertices in a graph. The method is attractive in that it has a linear complexity and can be easily parallelized. Based on an analysis of the convergence property, we propose a mutually reinforcing model to explain the intuition behind the strategy. The practical effectiveness of this measure is demonstrated through several combinatorial optimization problems on graphs and hypergraphs.

1   Introduction

Measuring the connectivity between two vertices in a graph is one of the central questions in many theoretical and applied areas in computer science. A variety of methods exist for this purpose, such as shortest path length, number of paths between vertices, maximum flow, and minimum vertex/edge cut/separator. In this paper, we discuss a strategy that measures the connectivity between the vertices that are located not very far from each other. It is a simple iterative process that is based on the edge weights and that takes into account the neighborhood information of each vertex. We will analyze some properties of the strategy and demonstrate how the connectivity estimations can be used in practice to improve several well-known algorithms. In particular, the connectivity measure can be used in algorithms with greedy steps where it is critical to choose an appropriate edge that is the "heaviest".

   Since the notion of connectivity is of practical significance, many algorithms have been developed to model it. In a random-walk approach [8, 18], the average first-passage time/cost and average commute time were used. A similarity measure between nodes of a graph integrating indirect paths, based on the matrix-forest theorem, was proposed in [4]. Approximation of a betweenness centrality in [1] makes this computationally expensive concept feasible. A convergence of the compatible relaxation [2] was measured in Algebraic Multigrid (AMG) schemes [3] in order to detect strong connections between fine and coarse points. A similarity method based on probabilistic interpretation of a diffusion was introduced in [14]. Our goal is to design a family of connectivity models and estimators that are fast and easy to implement and parallelize, and that can be applied on local parts of the data.

---

2   Definitions and Notations

Let $G = (V, E)$ denote a weighted simple connected graph, where $V = \{1, 2, ..., n\}$ is the set of nodes (vertices) and E is the set of edges. Denote by w_{ij} the non-negative weight of the undirected edge ij between nodes i and j; if $ij \notin E$, then $w_{ij} = 0$. Let $W = \{w_{ij}\}$ be the weighted adjacency matrix of G. The graph Laplacian matrix is defined as $L = D - W$, where D the diagonal matrix with diagonal elements $d_{ii} = \sum_j w_{ij}$. Correspondingly, the normalized Laplacian is $\mathcal{L} = D^{-1/2}LD^{-1/2}$. Denote by δ_i the degree of vertex i.

The proposed strategy is to initially assign a random value x_i to each vertex i and to update the value x_i one by one by a weighted combination of the value itself and the weighted average of i's neighbors. Then, after a few iterations, the absolute difference between x_i and x_j is an indicator of the coupling between i and j. This process is precisely stated in Algorithm 1 in the vector form, where we use superscripts such as $^{(k)}$ and $^{(k-1)}$ to distinguish successive iterates and use subscripts to mean vector entries. We define the *algebraic distance* (the discussed connectivity measure) between the vertex i and j, at the kth iteration, to be

$$s_{ij}^{(k)} := \left| x_i^{(k)} - x_j^{(k)} \right|. \tag{1}$$

With R initial vectors $x^{(0,r)}$, $r = 1, \ldots, R$, each vector is independently updated by using Algorithm 1, and the *extended p-normed algebraic distance* is defined as

$$\rho_{ij}^{(k)} := \left(\sum_{r=1}^{R} \left| x_i^{(k,r)} - x_j^{(k,r)} \right|^p \right)^{1/p}, \tag{2}$$

where the superscript $^{(k,r)}$ refers to the kth iteration on the rth initial random vector. For $p = \infty$, by convention, $\rho_{ij}^{(k)} = \max_{r=1}^{R} \left| x_i^{(k,r)} - x_j^{(k,r)} \right|$.

Algorithm 1 Computing algebraic distances for graphs

Input: Parameter ω, initial vector $x^{(0)}$
1: **for** $k = 1, 2, \ldots$ **do**
2:      $\tilde{x}_i^{(k)} \leftarrow \sum_j w_{ij} x_j^{(k-1)} / \sum_j w_{ij}, \quad \forall i.$
3:      $x^{(k)} \leftarrow (1 - \omega)x^{(k-1)} + \omega \tilde{x}^{(k)}$
4: **end for**

Conceptually, a *small* distance means a *strong* connection. The parameter ω is fixed to be $1/2$.

3   Historical Background and Motivation

The algebraic distance is motivated by the Bootstrap AMG (BAMG) method [3] for solving a symmetric positive definite system $Ax = b$. In this method, a Gauss-Seidel (GS) process is run on the system $Ax = 0$ in order to expose the slow-to-converge variables x_i to allow a better interpolation of the low residual errors. Recently, the extended ∞-normed algebraic distance was used as a component of an AMG based coarsening scheme for graph linear ordering problems [15]. In a multilevel graph coarsening framework, one of the most vital concerns is how to choose vertices

that are strongly coupled for merging (either in a strict or in a weighted sense), such that the coarse graph will faithfully represent the original one with respect to the given optimization problem [3]. Despite considerable empirical evidence of success in multilevel linear ordering algorithms, however, the concept of algebraic distance is still not well understood and has not been used widely in combinatorial optimization algorithms. This paper studies some properties of this relaxation process and interprets the algebraic distances under a mutually reinforcing environment model, where the neighborhood connectivity information governs the coupling of the vertices. In particular, two vertices are strongly connected if they are surrounded by similar neighborhoods. With this interpretation, the applications of this measure are no longer restricted to multilevel algorithms. Whenever the concept of vertex couplings is applicable, we can use the algebraic distances to measure the connectivity between vertices. We show a few such applications in this paper.

We note that an advantage of the proposed measure is its computational efficiency. As mentioned in the introduction, there exist other possible heuristics for measuring the connection strength between a pair of vertices, (e.g., the number of simple paths, the length of the shortest path, the commute time). However, these quantities are in general expensive to compute. For example, the problem of counting the number of simple paths connecting a pair of vertices is #P-complete [17], the Floyd-Warshall algorithm for computing all pairs shortest paths has an $O(n^3)$ time complexity, and to compute the commute times involves the pseudo-inverse of the Laplacian L. In contrast, since Algorithm 1 is essentially a Jacobi over-relaxation (JOR) process, its k iterations take only $O(km)$ time, where k is typically small, and m is the number of edges in the graph. This time cost is a significant reduction, especially for sparse graphs. Further, the JOR iterations are easy to parallelize because unlike other iterative processes such as GS or SOR, the update of an entry x_i does not require the most recent value of x_{i-1}. Thus, the proposed strategy has a strong potential for large-scale distributed computing.

4   Iterative Methods for Graph Laplacians

Algorithm 1, on which the definition of algebraic distances is based, is essentially the JOR method for solving the linear system

$$Lx = 0. \tag{3}$$

There being rich results for nonsingular systems, however, here the matrix L is singular, and thus we need to first study the convergence properties for this particular system. In this section, we establish some general results for the convergence of several classical iterative methods (including JOR) for (3). The special case for JOR automatically applies.

Standard iterative methods by matrix splitting for solving a linear system can be written in a general form

$$x^{(k+1)} = Hx^{(k)}, \qquad k = 0, 1, 2, \ldots, \tag{4}$$

where H is the iteration matrix. Let the Laplacian $L = D - W_L - W_U$, where W_L and W_U are the strict lower and upper triangular parts of W, respectively. Then the iteration matrices for Gauss Seidel, Jacobi, SOR, and JOR are, respectively,

$$H_{GS} = (D - W_L)^{-1} W_U, \qquad H_{SOR} = (D/\omega - W_L)^{-1} \left((1/\omega - 1)D + W_U \right),$$
$$H_{JAC} = D^{-1}(W_L + W_U), \qquad H_{JOR} = (D/\omega)^{-1} \left((1/\omega - 1)D + W_L + W_U \right).$$

We will use the notation H when the discussions are general or apply to all the iterative methods; we will add subscripts when an individual method is emphasized.

A matrix $A \in \mathbb{R}^{n \times n}$ is said to be *convergent* if $\lim_{k \to \infty} A^k$ exists.[1]

Theorem 1. *The iteration matrices H_{GS}, H_{SOR} (with $0 < \omega < 2$), and H_{JOR} (with $0 < \omega < 2/\rho(\mathcal{L})$) are convergent, with spectral radius $\rho(H) = 1$. The iteration matrix H_{JAC} is convergent if and only if none of the connected component of the graph is bipartite.*

We have the following result when H is convergent.

Theorem 2. *If the graph is connected, and the iteration matrix H for the linear system (3) is convergent, then the iterate $x^{(k)}$ converges to zero if the initial vector $x^{(0)} \in \mathrm{range}(I-H)$. Otherwise $x^{(k)}$ converges to a nonzero scalar multiple of $\mathbf{1}$ (a vector of all ones).*

Corollary 3. *Under the conditions of Theorem 2, the quantity $s_{ij}^{(k)}$ defined in (1) converges to zero for all i and j.*

To establish the rate of convergence, we make an additional mild assumption that H is diagonalizable. Let (σ_i, ϕ_i) denote the eigen-pairs of H, where the eigenvalues are labeled in the order

$$1 = \sigma_1 > |\sigma_2| \geq |\sigma_3| \geq \cdots \geq |\sigma_n|.$$

Corollary 4. *Under the conditions of Theorem 2, assume that H is diagonalizable with eigen-pairs (σ_i, ϕ_i) labeled in nonincreasing order of the magnitudes of the eigenvalues. Then the iterate $x^{(k)}$ approaches the limit in the order $O(|\sigma_2|^k)$, and the quantity $s_{ij}^{(k)}$ defined in (1) approaches zero in the same order.*

The results of Corollaries 3 and 4 seem to suggest that the definition of the algebraic distance as a measure of the strength of connection is inappropriate. However, we are actually interested in comparing the relative magnitudes of $s_{ij}^{(k)}$ for different (i,j) pairs. In other words, a concurrent scaling of the quantity $s_{ij}^{(k)}$ for all i and j will not compromise the measure. To this end, we consider the quantity

$$\hat{s}_{ij}^{(k)} := s_{ij}^{(k)}/\sigma_2^k. \tag{5}$$

We have the following result.

Theorem 5. *Under the conditions of Corollary 4, let the initial vector $x^{(0)}$ be expanded in the eigenbasis of H as $x^{(0)} = a_1\phi_1 + a_2\phi_2 + \cdots + a_n\phi_n$.*

(i) *If $\sigma_2 = \sigma_3 = \cdots = \sigma_t$ and $|\sigma_t| > |\sigma_{t+1}|$ for some $t \geq 2$, and if $a_2, \ldots, a_t$ are not all zero, then the quantity $\hat{s}_{ij}^{(k)}$ defined in (5) approaches the limit $|(e_i - e_j)^T \xi|$ in the order $O\left(|\sigma_{t+1}/\sigma_t|^k\right)$, where ξ is an eigenvector corresponding to the eigenvalue σ_2 (with multiplicity $t-1$).*

(ii) *If $|\sigma_2| = |\sigma_3| = \cdots = |\sigma_t| > |\sigma_{t+1}|$ for some $t \geq 3$, where $\sigma_2, \ldots, \sigma_t$ are not all the same, $a_2, \ldots, a_t$ are not all zero, and if there exists an integer m such that $(\sigma_\ell/\sigma_2)^m = 1$ for $\ell = 3, \ldots, t$, then the p-th subsequence $\{\hat{s}_{ij}^{(mk+p)}\}_{k=0,1,2,\ldots}$ approaches the limit $|(e_i - e_j)^T \eta_p|$ in the order $O\left(|\sigma_{t+1}/\sigma_t|^{mk}\right)$, where $\eta_p = a_2\phi_2 + a_3(\sigma_3/\sigma_2)^p\phi_3 + \cdots + a_t(\sigma_t/\sigma_2)^p\phi_t$ for $p = 0, 1, \ldots, m-1$.*

---

[1] Note that some authors (e.g., [11]) use the term *convergent* for a matrix A where the limit A^k is zero. However, the interesting case in this paper is that the limit is nonzero. Thus, we make a broader inclusion in the definition here.

5 Jacobi Over-Relaxations

In this section, we discuss the implications of the theorems in the previous section for the JOR iterations. Specifically, the eigenvalues and vectors of the iteration matrix H_{JOR} are closely related to those of the matrix pencil (L, D). Because of the distributions of the eigenvalues, the convergence of Algorithm 1 often is slow. Hence, we also study the behavior of the iterations at an early stage.

5.1 Algebraic Distances at the Limit

An immediate result is that H_{JOR} is diagonalizable and all the eigenvalues of H_{JOR} are real, since

$$H_{JOR}\ \phi_i = \sigma_i \phi_i \iff L\phi_i = \frac{1 - \sigma_i}{\omega} D\phi_i.$$

This equivalence implies that if μ_j is an eigenvalue of (L, D), then $\mu_j = (1 - \sigma_i)/\omega$ for some i. In general, we may not have $\mu_i = (1 - \sigma_i)/\omega$ for all i, since the eigenvalues of H_{JOR} are sorted in decreasing order of their magnitudes, whereas the eigenvalues of (L, D) are sorted in their natural order. In particular, depending on the value of ω, σ_2 can be either $1 - \omega\mu_2$ or $1 - \omega\mu_n$, and there are more possibilities for σ_3. Enumerating all the possible cases, we have the following theorem as a corollary of case (i) of Theorem 5.

Theorem 6. *Given a connected graph, let $(\mu_i, \hat{v}_i)$ be the eigen-pairs of the matrix pencil (L, D), labeled in nondecreasing order of the eigenvalues, and assume that $\mu_2 \neq \mu_3 \neq \mu_{n-1} \neq \mu_n$. Unless $\omega = 2/(\mu_2 + \mu_n)$, the quantity $\hat{s}_{ij}^{(k)}$ defined in (5) will always converge to a limit $|(e_i - e_j)^T \xi|$ in the order $O(\theta^k)$, for some ξ and $0 < \theta < 1$. If $0 < \omega < 2/(\mu_2 + \mu_n)$, then $\xi \in \mathrm{span}\{\hat{v}_2\}$; otherwise (if $2/(\mu_2 + \mu_n) < \omega < 2/\mu_n$), $\xi \in \mathrm{span}\{\hat{v}_n\}$.*

A graphical illustration of the dependence of θ on ω is shown in Figure 1.
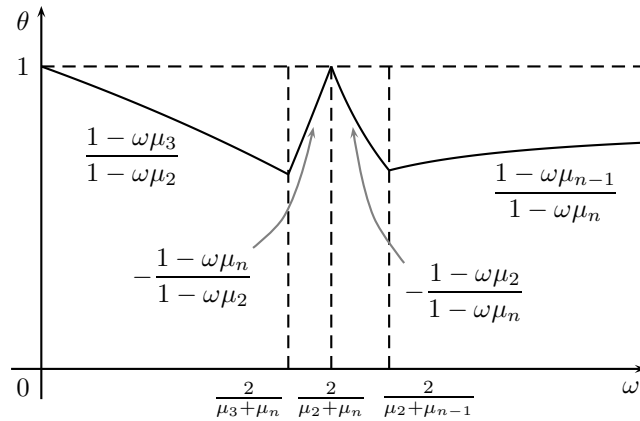

Figure 1: The θ as a function of ω. Note that the value $2/\mu_n$ can be less than, equal to, or greater than $2/(\mu_2 + \mu_{n-1})$.

Theorem 6 shows two possible limits depending on the value of ω. We can derive some upper/lower bounds for the cutting point $2/(\mu_2 + \mu_n)$ and estimate which of the two cases in Theorem 6 is applied (will be given in full version of the paper). For example, if the graph is not

complete, we have $2/(\mu_2 + \mu_n) \geq 2/3$, since in such a case $\mu_2 \leq 1$. In practice, we deal with sparse graphs and we set $\omega = 1/2$. Therefore, $\hat{s}_{ij}^{(k)}$ always converges to $|(e_i - e_j)^T \xi|$ with $\xi \in \text{span}\{\hat{v}_2\}$.

5.2 Algebraic Distances at Early Iterations

Sometimes, even the optimal θ is so close to one that the theoretical convergence of $\hat{s}_{ij}^{(k)}$ is of little practical use—it takes an enormous number of steps before it gets close enough to the limit. (As observed for some real-life graphs, the smallest possible θ in Figure 1 can be as high as 0.999.) However, an interesting phenomenon is that in practice $x^{(k)}$ soon becomes "stable"; that is, the two iterates $x^{(k+1)}$ and $x^{(k)}$ are almost parallel even when k is small.

Theorem 7. *Given a graph, let $(\mu_i, \hat{v}_i)$ be the eigen-pairs of the matrix pencil (L, D), labeled in nondecreasing order of the eigenvalues. Denote $\hat{V} = [\hat{v}_1, \ldots, \hat{v}_n]$. Let $x^{(0)}$ be the initial vector, and let $a = \hat{V}^{-1} x^{(0)}$ with $a_1 \neq 0$. If the following two conditions are satisfied:*

$$1 - \omega\mu_n \geq 0, \tag{6a}$$

$$f_k := \frac{\alpha r_k^{2k}(1 - r_k)^2}{1 + \alpha r_k^{2k}(1 + r_k)^2} \leq \frac{1}{\kappa}, \tag{6b}$$

where $\alpha = \left(\sum_{i \neq 1} a_i^2\right) / \left(4a_1^2\right)$, r_k is the unique root of the equation

$$2\alpha r^{2k+1}(1 + r) = k - (k + 1)r$$

on the interval $[0, 1]$, and κ is the condition number of D, then

$$1 - \left\langle \frac{x^{(k)}}{\|x^{(k)}\|}, \frac{x^{(k+1)}}{\|x^{(k+1)}\|} \right\rangle^2 \leq \frac{4\kappa f_k}{(1 + \kappa f_k)^2}. \tag{7}$$

Note that when $\omega = 1/2$, which is the relaxation parameter we use, (6a) is satisfied, since the spectral radius of (L, D), μ_n, is less than or equal to 2. Also note the condition number κ of D. For many graphs arising from application areas such as VLSI design and finite element meshes, if the graph edges have a uniform weight equal to one, then d_{ii} is the degree of a vertex. Thus, κ will not be large. The bound (7) serves as a reference for estimating how close to parallel two iterates are. In practice, for all the graphs we have experimented with, when k is equal to 30 or 50, the quantity on the left-hand side of (7) has dropped to the order of 10^{-4}. Note that $\sin^2(\pi/180) = 3.05 \times 10^{-4}$.

6 Mutually Reinforcing Model

The local structure of a graph and the edge weights are two factors that mutually govern the strength of the connection between a pair of vertices. We present here a model that incorporates both of the factors for quantitatively evaluating the vertex connectivity. Consider a mutually reinforcing environment, where entities are influenced by their neighbors. Intuitively, for an abstract property that is characteristic in such an environment, a part of the property value for an entity should be a weighted average of the influences from its neighbors in some way. Two entities are said to be close, or similar, if they are placed in two similar environments, or, consequently, their property values are close. If we consider the graph itself as an integral environment and vertices as individual

entities each of which is surrounded by a neighborhood (the neighboring vertices), then two vertices are strongly coupled if they have similar values for an afore mentioned abstract property. Let each vertex i be associated with a real number x_i. Except for a μ portion of itself, i is influenced by its neighbors, which is quantitatively a weighted average:

$$x_i = \mu x_i + \sum_{j \sim i} p_{ij} x_j, \tag{8}$$

where $j \sim i$ means j is a neighbor of i. Here, the portion $0 \leq \mu \leq 1$ is an indicator of how strongly an environment will act on a vertex. When μ tends to zero, the neighborhood plays a major role, whereas when μ tends to one, a vertex is so stubborn that its neighbors cannot have a strong impact on it. The coefficient μ does not need to be explicitly specified; it is an internal property of the entire environment (i.e., the graph). For such a mutually reinforcing environment, a small μ is more desired. The weight of the influence by a neighbor j, p_{ij}, should be non-negative, and all the p_{ij} related to the same i should sum up to one. The weight p_{ij} reflects how strong a neighbor can influence i, and therefore a natural choice is $p_{ij} = w_{ij}/\sum_j w_{ij}$. Thus, (8) is formally written in the following way

$$x_i = \mu x_i + \sum_j \frac{w_{ij}}{d_{ii}} x_j \qquad (0 \leq \mu \leq 1). \tag{9}$$

Equivalently, in the matrix form, it is

$$x = \mu x + D^{-1} W x. \tag{10}$$

The coupling of two vertices i and j is measured by $|x_i - x_j|$. A small value means a strong connection, which equivalently means that their neighborhoods have a similar influence on the two vertices.

From (10) we see that x is an eigenvector of the matrix pencil (L, D) and that μ is its corresponding eigenvalue. Unless the graph is complete, we have at least two sets of x and μ that satisfy this system. In the first set, μ is zero, and x is a nonzero scalar multiple of $\mathbf{1}$. In this case, since $\mu = 0$, the value of each vertex is entirely determined by its neighbors. This would have been the most desirable situation for a mutually reinforcing environment because it means that every entity is influenced only by its neighborhood. However, this situation leads to the result that every entity is the same (x_i constant for all i), and therefore no discriminating power is presented. In the second set, μ is equal to μ_2, the second smallest eigenvalue of (L, D), and $x = \hat{v}_2$. When the graph is not complete, $\mu_2 \leq 1$. Indeed, frequently μ_2 is close to zero in practice. This is naturally a desirable solution for our problem: the neighborhood has a strong impact on a vertex, and vertices have different values such that the strengths of the connectivity for different vertex pairs can be distinguished.

We have proved that soon after the iterations start, two successive iterates are getting close to parallel. In practice, we observe that when this situation happens, $x^{(k)}$ and $x^{(k+1)}$ form an acute angle and $\left\| H_{JOR} \, x^{(k)} \right\| \approx (1 - \omega \mu_2) \left\| x^{(k)} \right\|$. (This makes sense because $\| H_{JOR} \, \hat{v}_2 \| = 1 - \omega \mu_2$.) Denote $\hat{x}^{(k)} = x^{(k)} / \left\| x^{(k)} \right\|$. Then, we have

$$\hat{x}^{(k)} \approx \mu_2 \hat{x}^{(k)} + D^{-1} W \hat{x}^{(k)}.$$

This means that the (normalized) iterate $x^{(k)}$, when close to parallel to the next iterate, approximately satisfies the model (9), with $\mu = \mu_2$. In other words, the algebraic distance $s_{ij}^{(k)} =$

7

$\left| x_i^{(k)} - x_j^{(k)} \right|$, computed from $x^{(k)}$, approximately measures the connection strength between i and j in our model.

We remark that for a small k, the iterate $x^{(k)}$ can be quite different from its limit $\hat{v}_2$, and for different initializations, $x^{(k)}$ will be different. However, they all satisfy or approximately satisfy the mutually reinforcing model (9). This gives us the flexibility, yet not the arbitrariness, to estimate the connectivity for different vertex pairs. Readers may question why an iterate $x^{(k)}$ is preferred over the eigenvector $\hat{v}_2$ as the measure. A major reason is that the JOR method with a few number of iterations is computationally much less expensive than computing an eigenvector, even when the matrix L (or pencil (L, D)) is sparse. Solving a large scale sparse eigenvalue problem for a symmetric matrix, say, using the Lanczos method [13, 16, 10], involves frequent convergence tests, each of which needs to solve an eigenvalue subproblem for a tridiagonal matrix. On the other hand, inside each JOR iteration is nothing but weighted averages. Therefore, the JOR process is particularly inexpensive compared with computing an eigenvector. Besides, the simplicity of Algorithm 1 makes it particularly attractive, and thus it is advocated as *the* algorithm for the proposed measure in this paper.

7 Applications

In this section, we demonstrate how the algebraic distance can be used in practice. For this purpose, we have chosen four problems: maximum weighted matching, maximum independent set and the minimum τ-partitioning of graphs and hypergraphs. In all these cases, fast existing baseline algorithms were modified by taking into account the algebraic distance instead of the original graph edge weights. The experimental graphs were of different sizes ($|E|$ was between 10^3 and 10^7) and have been selected from the UFL database of real-life matrices [6]. Because of page limitation, we present only two applications here. The other two applications will be included in the full paper.

7.1 Maximum Weighted Matching

A matching, M, of G is a subset of E such that no vertex in V is incident to more than one edge in M. A matching M is said to be *maximum* if, for any other matching M', $|M| \geq |M'|$. Similarly, a matching M is said to be *maximum weighted* if, for any other matching M', $w(M) \geq w(M')$, where $w(M) = \sum_{ij \in M} w_{ij}$.

In many practical applications two well-known 2-approximations are used. One is a textbook greedy algorithm [5] for maximum weighted matching; the other is its improved version which is based on the path growing principle. Both algorithms are presented in [7].

Based on these algorithms and the algebraic distances, we designed two heuristics for the maximum weighted matching. In both algorithms, there exists a greedy step in which the next heaviest edge has to be chosen. The criterion for choosing an edge was changed according to the following heuristic observation: a better matching can be found in a dense graph with less effort than in a sparse graph. According to this observation, we give preference to matching two nodes that are not connected well with other nodes from their neighborhood. The preprocessing for greedy algorithms is presented in Algorithm 2.

The resulting values s'_{ij} are used in greedy choice steps (and sorting if applicable) instead of graph weights. The experimental results of the comparison are presented in Figure 2a as ratios between the textbook greedy matching *with* preprocessing and the same algorithm *without* pre-

8

Algorithm 2 Preprocessing for greedy algorithm for maximum matching
---
Input: Graph G
 1: For all edges $ij \in E$ calculate $\rho_{ij}^{(k)}$ for some k, R and p
 2: For all nodes $i \in V$ define $a_i = \sum_{ij \in E} 1/\rho_{ij}^{(k)}$
 3: For all edges $ij \in E$ define $s'_{ij} = a_i/\delta_i + a_j/\delta_j$
---

processing. Almost identical results were obtained by improving a greedy path growing algorithm from [7]. These particular results were obtained with $k = 20$, $R = 10$, and $p = \infty$. However, results of almost the same quality have been obtained with many different combinations of $R \geq 5$, $10 \leq k \leq 100$, and $p = 1, 2$.

7.2   Hypergraph Partitioning

Extending the algebraic distance for hypergraphs is the next step in our future research. Here we present some preliminary results of defining the algebraic distances on hypergraphs and experimenting with them on the hypergraph partitioning problem.

We define a hypergraph $\mathcal{H}$ as a pair $\mathcal{H} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of hyperedges. Each $h \in \mathcal{H}$ is a subset of $\mathcal{V}$. A hypergraph τ-partitioning is a well-known NP-hard problem (see [9] for its graph version). The goal of the problem is to find a partitioning of $\mathcal{V}$ into a family of τ disjoint nonempty subsets $(\pi_p)_{1 \leq p \leq \tau}$, while enforcing the following:

$$\text{minimize} \sum_{\substack{h \in \mathcal{E} \text{ s.t. } \exists i,j \in h \text{ and} \\ i \in \pi_p \Rightarrow j \notin \pi_p}} w_h \tag{11}$$

$$\text{such that } \forall p \in [1, \tau], |\pi_p| \leq (1 + \alpha) \cdot \frac{|V|}{\tau} \; ,$$

where α is a given *imbalance factor*. In this paper, we refer to the problem with $\tau = 2$.

HMetis2 [12] is one of the fastest and most successful modern solvers for partitioning problems. We use it as a black-box solver and define its extension HMetis2+ in Algorithm 3. First, we extend the algebraic distance for hypergraphs using its bipartite model. In particular, we create $G = (V, E)$ with $V = \mathcal{V} \bigcup \mathcal{E}$ and $ij \in E$ if $i \in \mathcal{V}$ appears in edge $j \in \mathcal{E}$. The edge weights are preserved with no change. Second, HMetis2 is applied on hypergraph with new hyperedge weights, namely inverse (*small* algebraic distance replaces *heavy* edge weight) algebraic distances.

The numerical results of comparing HMetis2 and HMetis2+ are presented in Figure 2b. Since HMetis2 is a multilevel algorithm, a more correct way to apply the algebraic distances is to use them at all levels as it was demonstrated in [15]. Even in these preliminary results, however, one can easily see that a significant improvement can be obtained by reinforcing a black-box algorithm with algebraic distances.

8   Conclusion

In this paper, we proposed a simple iterative algorithm for measuring the connectivity between graph vertices. We presented a convergence analysis of the algorithm, and interpreted the proposed measure by using a mutually reinforcing model. Empirical results show that the proposed measure is effective in several combinatorial optimization problems.

Algorithm 3 HMetis2+ : First eight lines—algebraic distance related preprocessing

---

Input: Hypergraph $\mathcal{H}$, $k = 20$, $R = 10$
1: $G = (V, E) \leftarrow$ bipartite graph model
2: Create R initial vectors $x^{(0,r)}$
3: **for** $r = 1, 2, \ldots, R$ **do**
4:     **for** $m = 1, 2, \ldots, k$ **do**
5:         $x_i^{(m,r)} \leftarrow \sum_j w_{ij} x_j^{(m-1,r)} / \sum_j w_{ij}, \quad \forall i.$
6:     **end for**
7: **end for**
8: **return** modified edge weights $s_h^{(k)} \leftarrow \sum_r \max_{i,j \in h} \left| x_i^{(k,r)} - x_j^{(k,r)} \right|, \quad \forall h \in \mathcal{E}.$

Input: Hypergraph $\mathcal{H}$ with hyperedge weights $1/s_h^{(k)}$
1: $C \leftarrow$ hyperedge cut obtained by HMetis2 on $\mathcal{H}$ with the modified edge weights
2: **return** cost of C with original edge weights

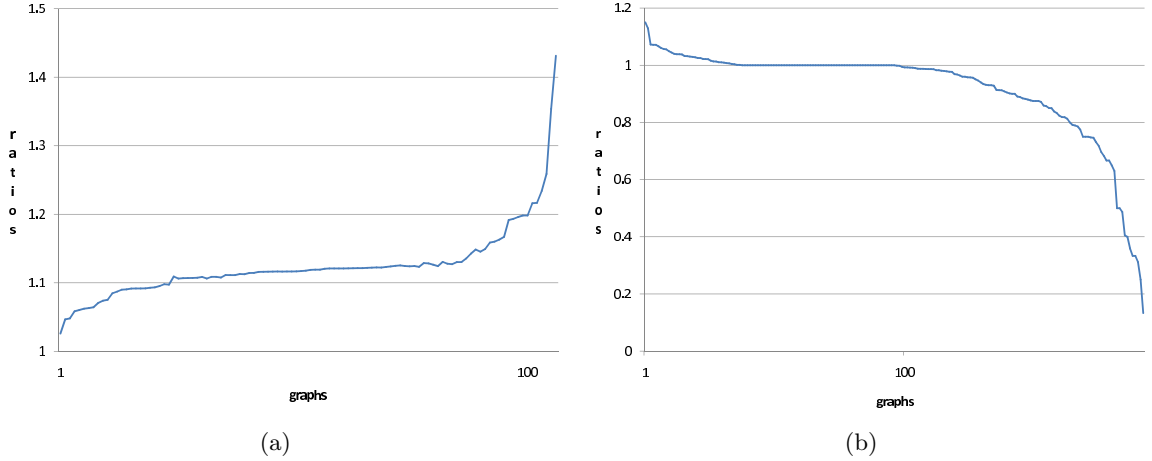(a)   (b)

Figure 2: **(a)** Comparison of greedy algorithms for matching with and without algebraic distance preprocessing. Each point corresponds to the average of ratios between matching sizes costs produced by the greedy algorithm with preprocessing and the same algorithm without preprocessing for one graph. The average was calculated over 20 different executions with different random initial vectors. The total number of graphs is 100. **(b)** Comparison of HMetis2 and HMetis2+ for hypergraph 2-partitioning. Each point corresponds to the average of ratios between cut costs produced by HMetis2 and HMetis2+ for one graph. The average was calculated over 20 different executions with different random seeds. The total number of graphs is 200.

References

[1] David Bader, Shiva Kintali, Kamesh Madduri, and Milena Mihail. Approximating betweenness centrality. *Algorithms and Models for the Web-Graph*, pages 124–137, 2007.

[2] A. Brandt. General highly accurate algebraic coarsening. *Electronic Trans. Num. Anal. 10 (2000) 1-20*, 2000.

[3] A. Brandt and D. Ron. Chapter 1 : Multigrid solvers and multilevel optimization strategies. In J. Cong and J. R. Shinnerl, editors, *Multilevel Optimization and VLSICAD*. Kluwer, 2003.

[4] P. Chebotarev and E. Shamis. On proximity measures for graph vertices. *Automation and Remote Control*, 59(10):1443–1459, 1998.

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. McGraw-Hill, 2nd edition, 2001.

[6] T. Davis. University of florida sparse matrix collection. *NA Digest*, 97(23), 1997.

[7] Doratha E. Drake and Stefan Hougardy. A simple approximation algorithm for the weighted matching problem. *Inf. Process. Lett.*, 85(4):211–213, 2003.

[8] Francois Fouss, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):355–369, 2007.

[9] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1:237–267, 1976.

[10] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[11] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[12] G. Karypis and V. Kumar. METIS *A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*. University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN, September 1998.

[13] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linea r differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45:255–282, 1950.

[14] Boaz Nadler, Stéphane Lafon, Ronald R. Coifman, and Ioannis G. Kevrekidis. Diffusion maps, spectral clustering and eigenfunctions of Fokker-Planck operators. In *Advances in Neural Information Processing Systems 18*, pages 955–962. MIT Press, 2005.

[15] Dorit Ron, Ilya Safro, and Achi Brandt. Relaxation based coarsening for combinatorial optimization problems. *submitted*, 2009.

[16] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Halstead Press, 1992.

[17] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[18] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications.* Structural Analysis in the Social Sciences. Cambridge University Press, 1994.